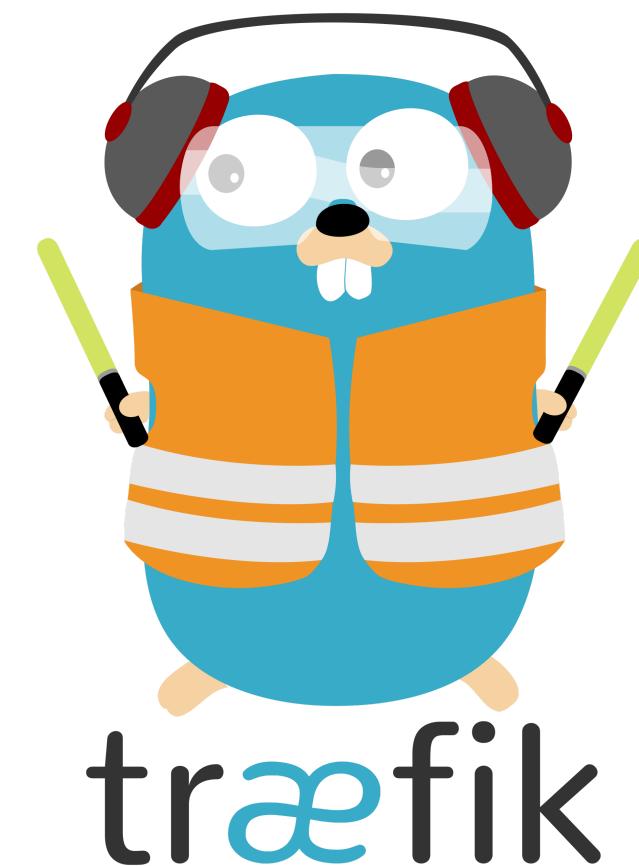


A Journey With Traefik 2.0 In Kubernetes



San Francisco Kubernetes Meetup - Wednesday 18th of
September 2019

How To Use These Slides?

- **Browse the slides:** Use the arrows
 - Change chapter: Left/Right arrows
 - Next or previous slide: Top and bottom arrows
- **Overview of the slides:** keyboard's shortcut "o"
- **Speaker mode (and notes):** keyboard's shortcut "s"

Whoami

Emile vauge

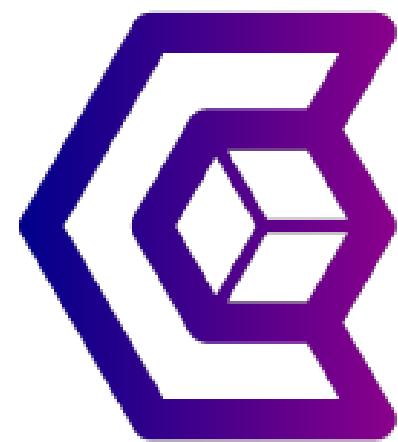
- 🚧 Developer
- Creator of **Træfik**, Founder of **Containous**
- 🐦 @emilevauge
- 🐱 emilevauge



Containous

<https://containo.us>

- We Believe in Open Source
- We Deliver Traefik and Traefik Enterprise Edition
- Commercial Support
- 30 people distributed, 90% tech

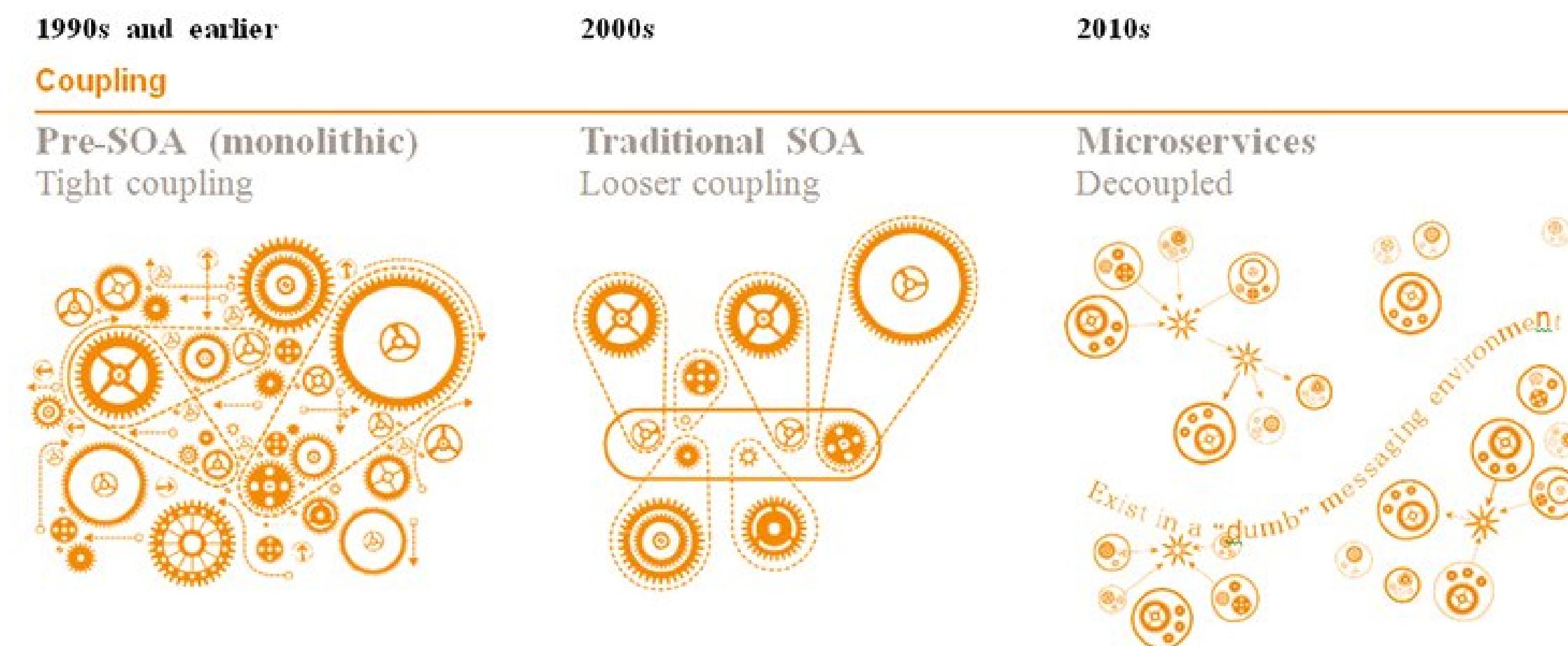


Why Traefik?



Why, Mr Anderson?

Evolution Of Software Design



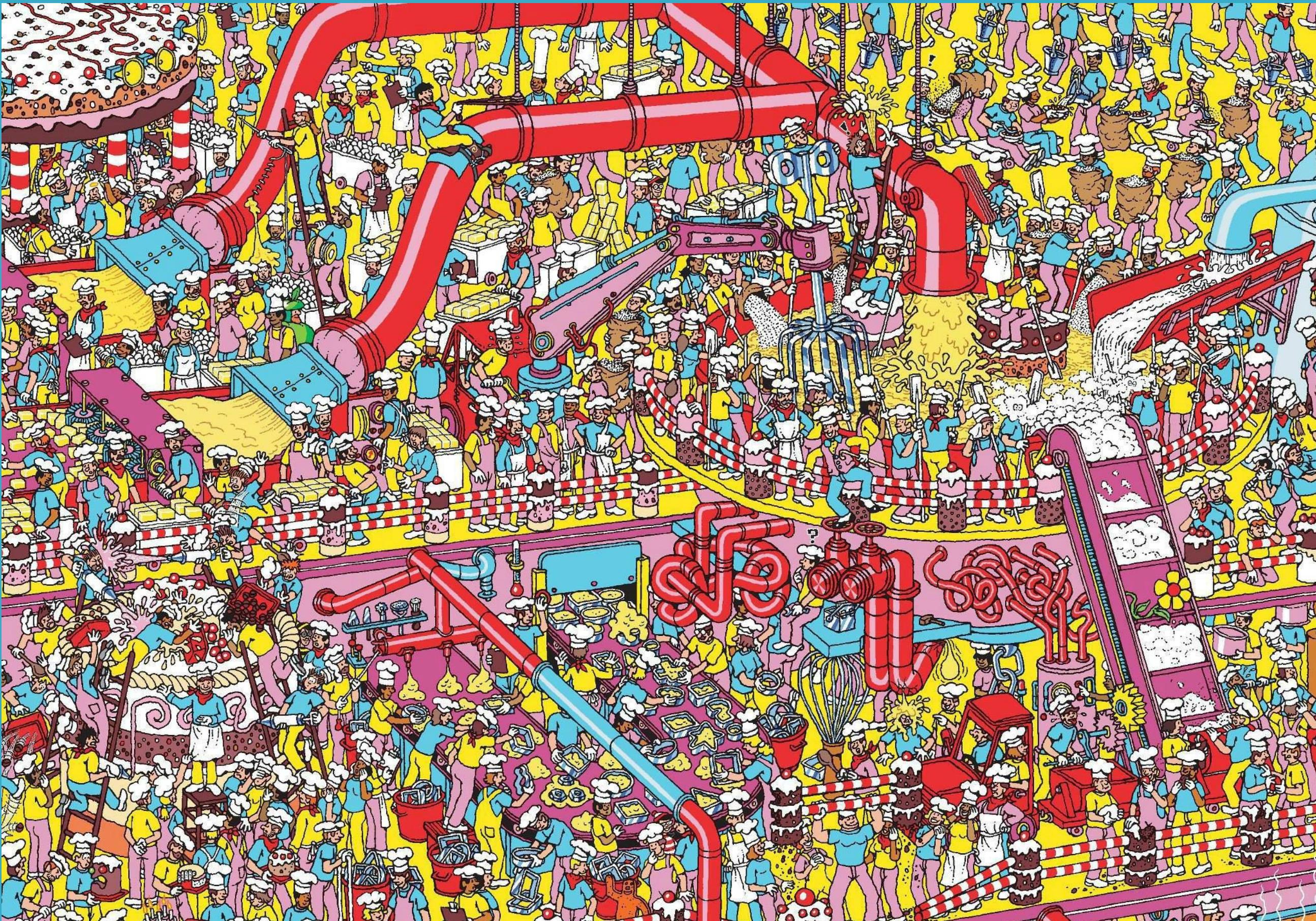
The Premise Of Microservices...



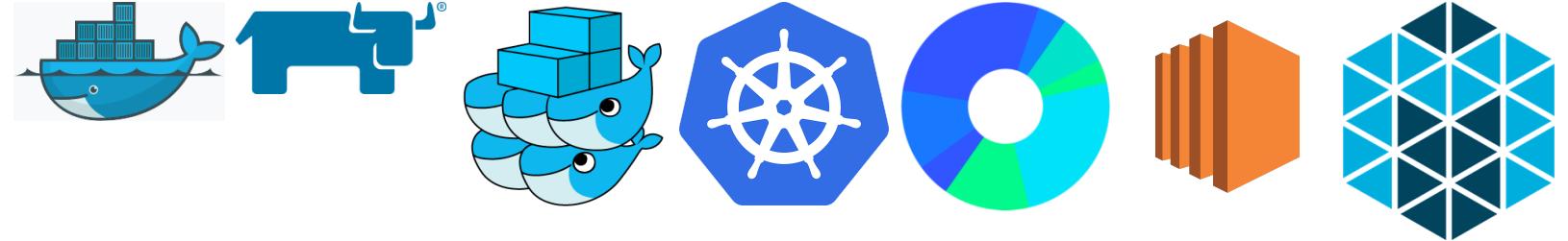
...And What Happens

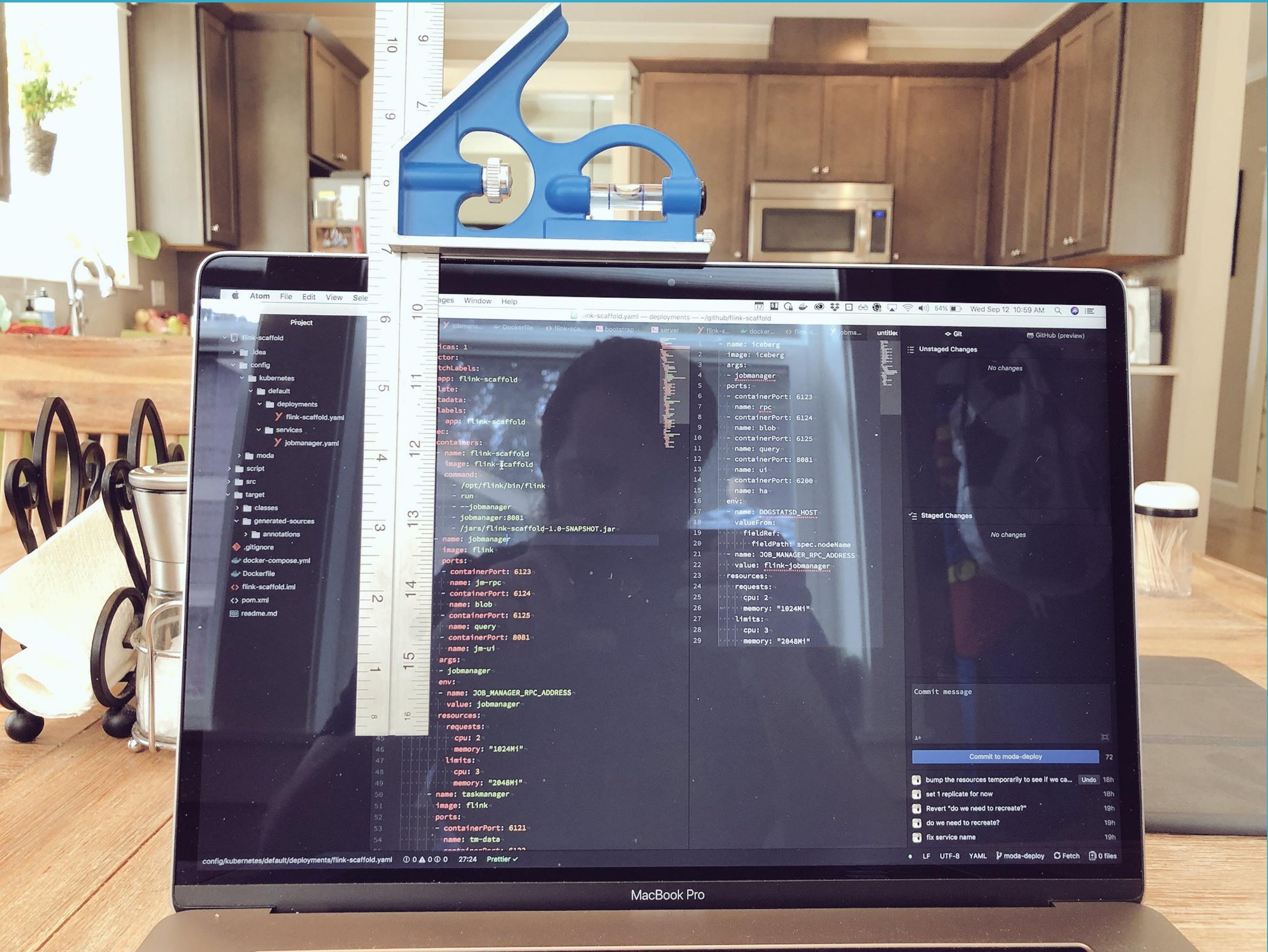


Where's My Service?



Tools Of The Trade





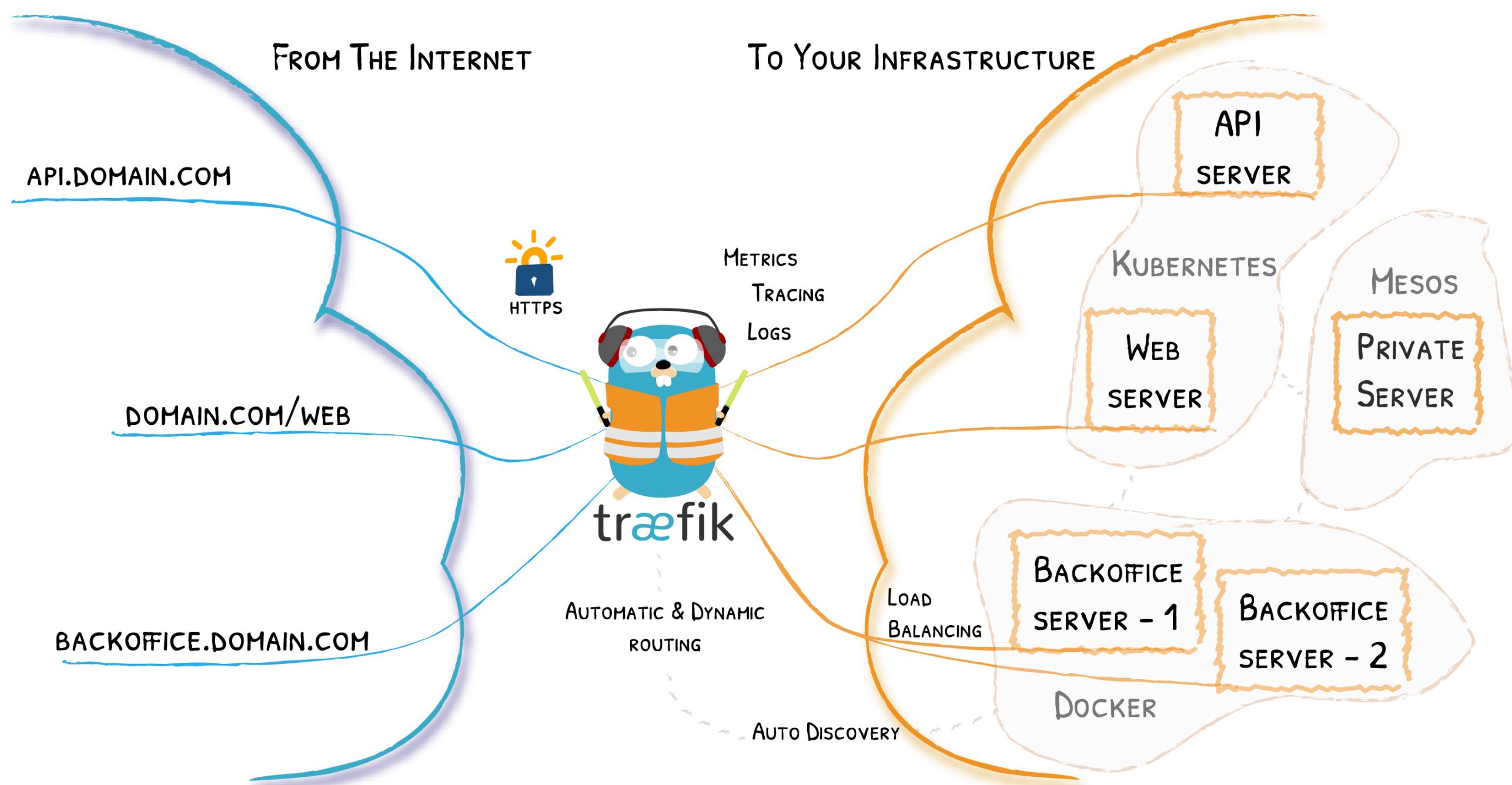
Source: <https://twitter.com/Caged/status/1039937162769096704>

What If I Told You?



That You Don't Have to Write This Configuration File...?

Here Comes Traefik!



Traefik Project

-  <https://github.com/containous/traefik>
- MIT License
- Written in Go
- 24,000+  1B+  400+ 
- Created in 2015, 4Y 
- Current stable branch: v2 . 0

BACK toTRAEFIK 2.0

Part →

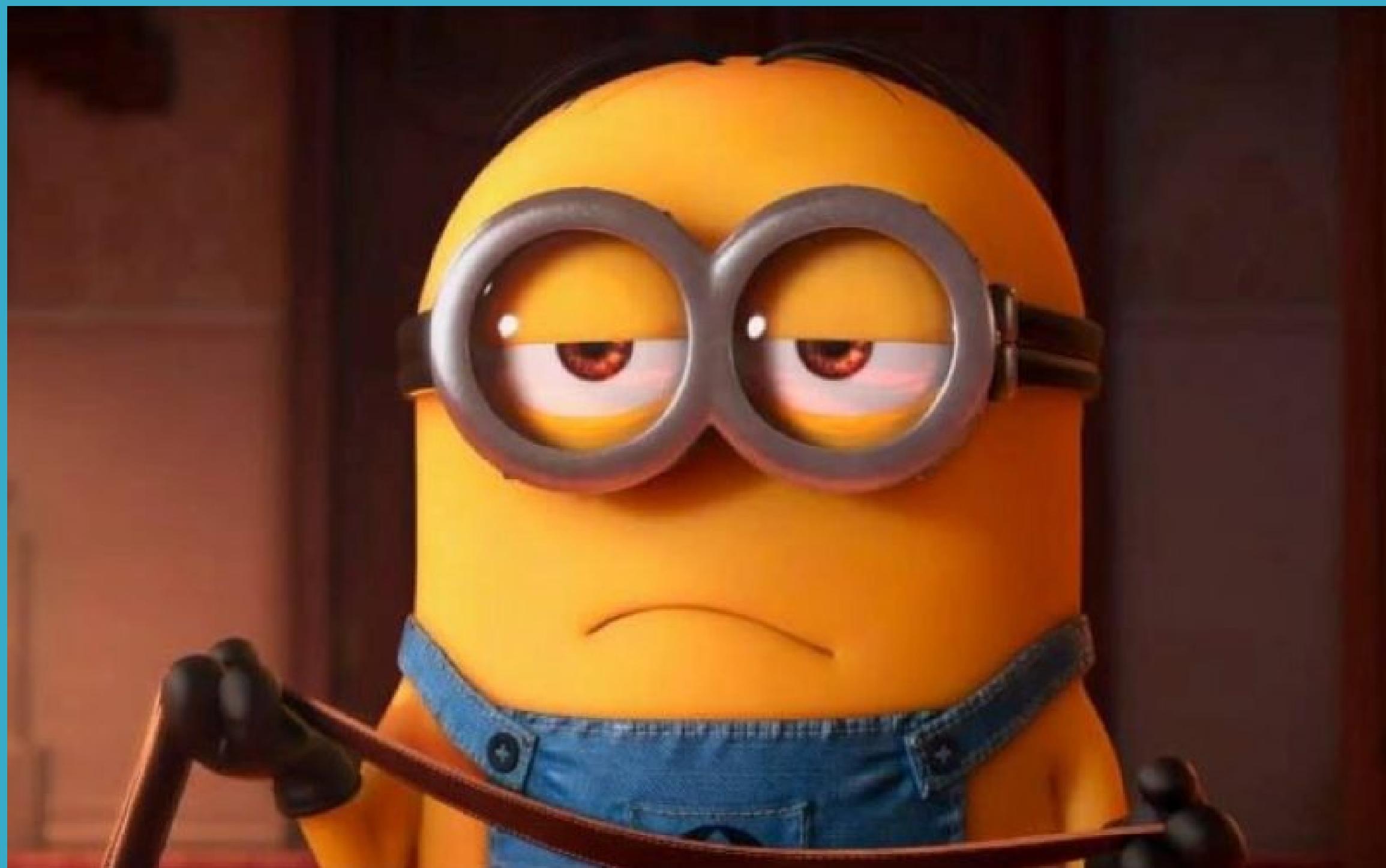


Traefik 2.0 Quick Overview

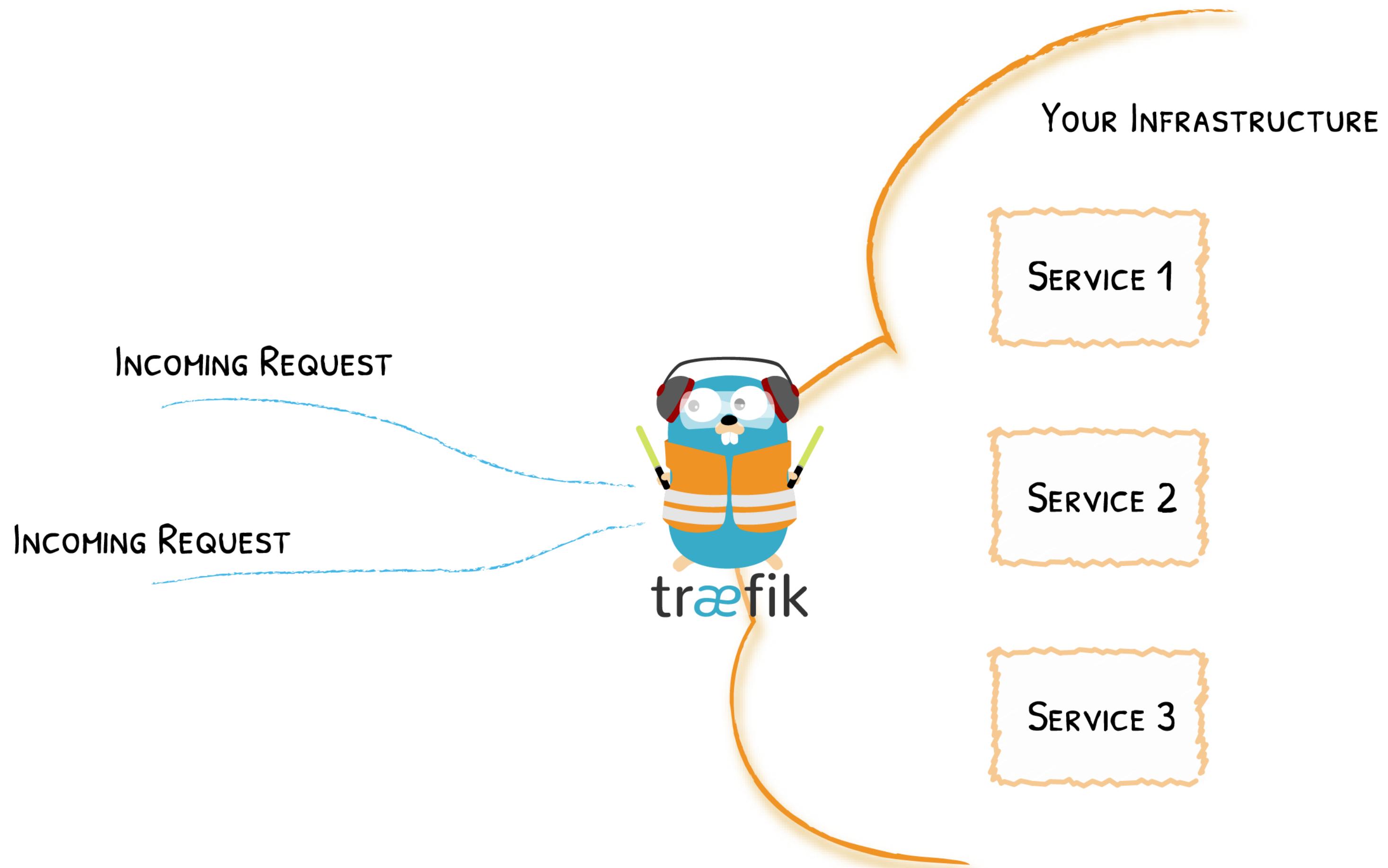
- Revamped Documentation
- Clarified Concepts
- Expressive Routing Rule Syntax
- Middlewares
- TCP Support
- Canary / Mirroring
- And so Much More...

Learn more on the blog post

Traefik (V2.0) Core Concepts

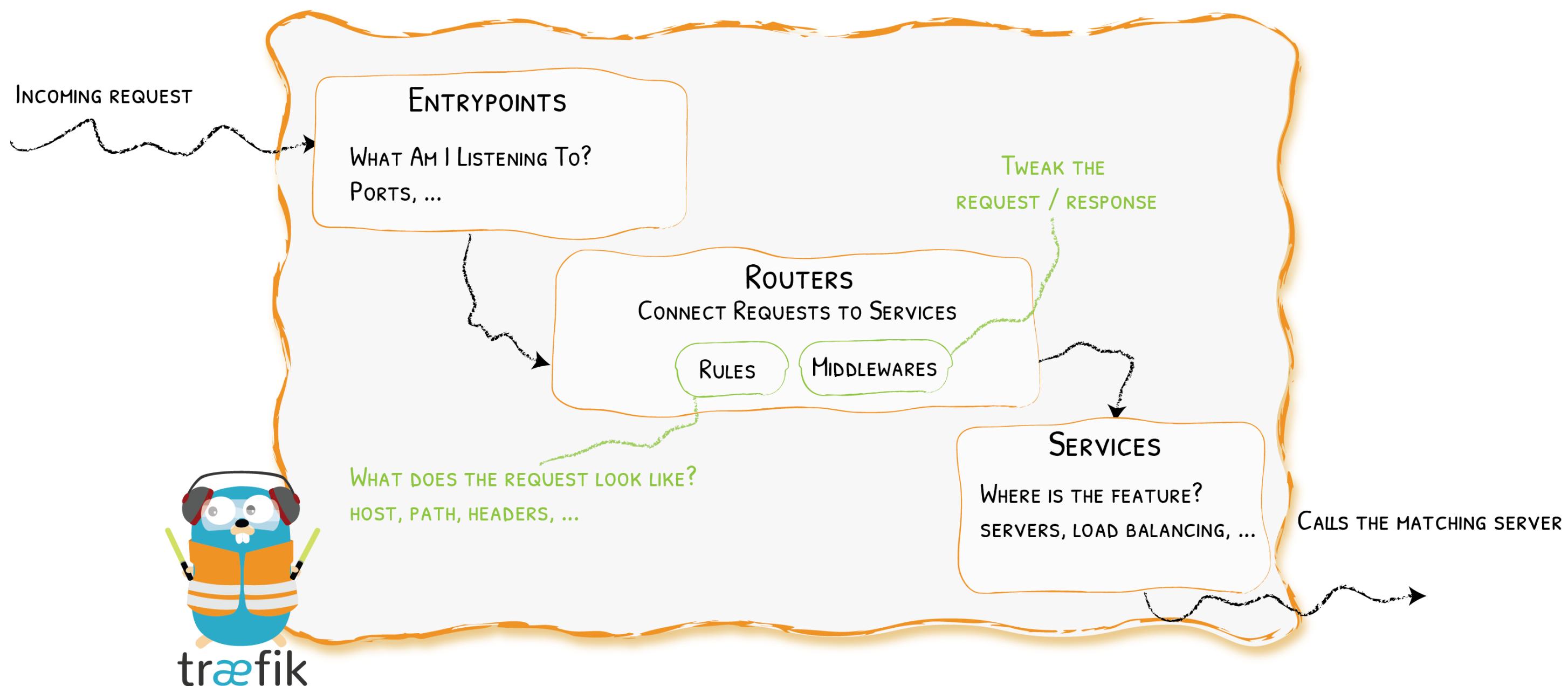


Traefik Is An Edge Router

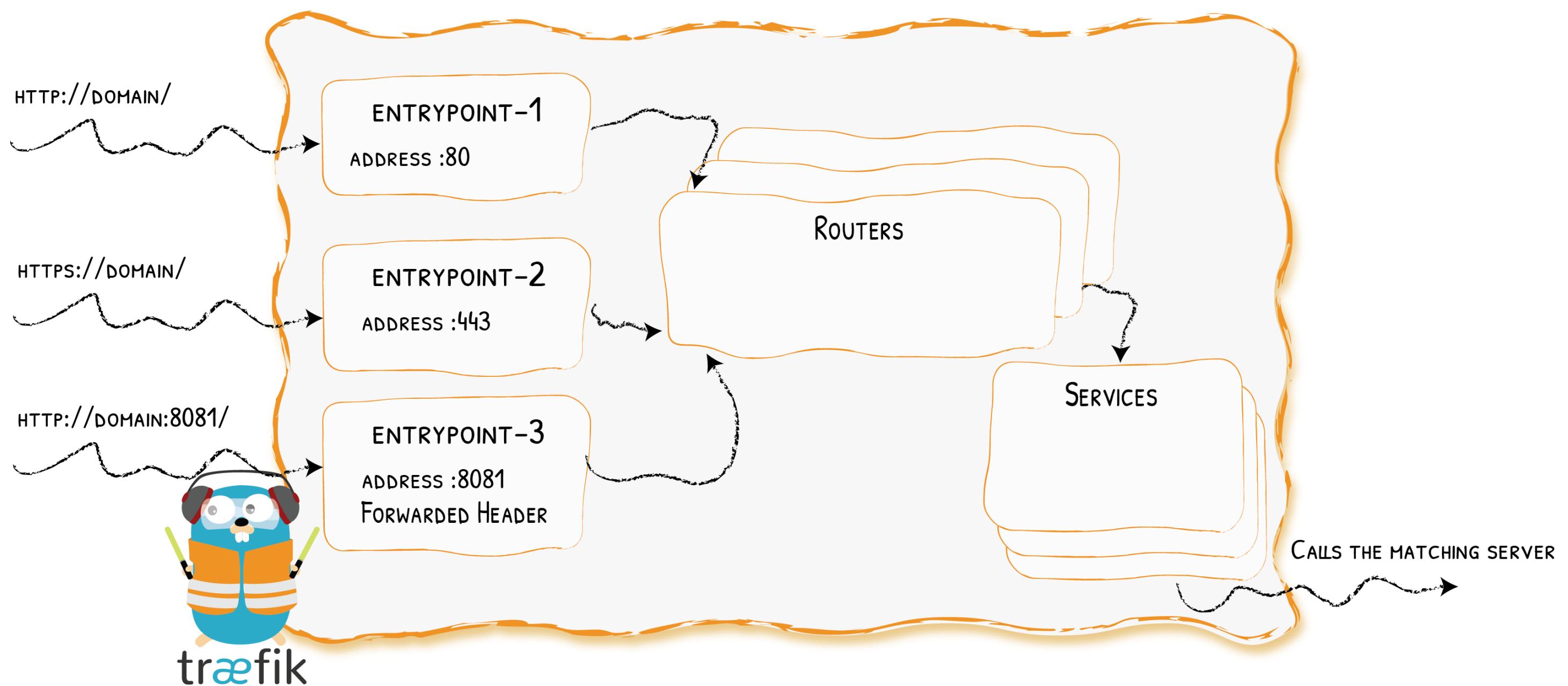


Traefik Dynamically Discovers Services

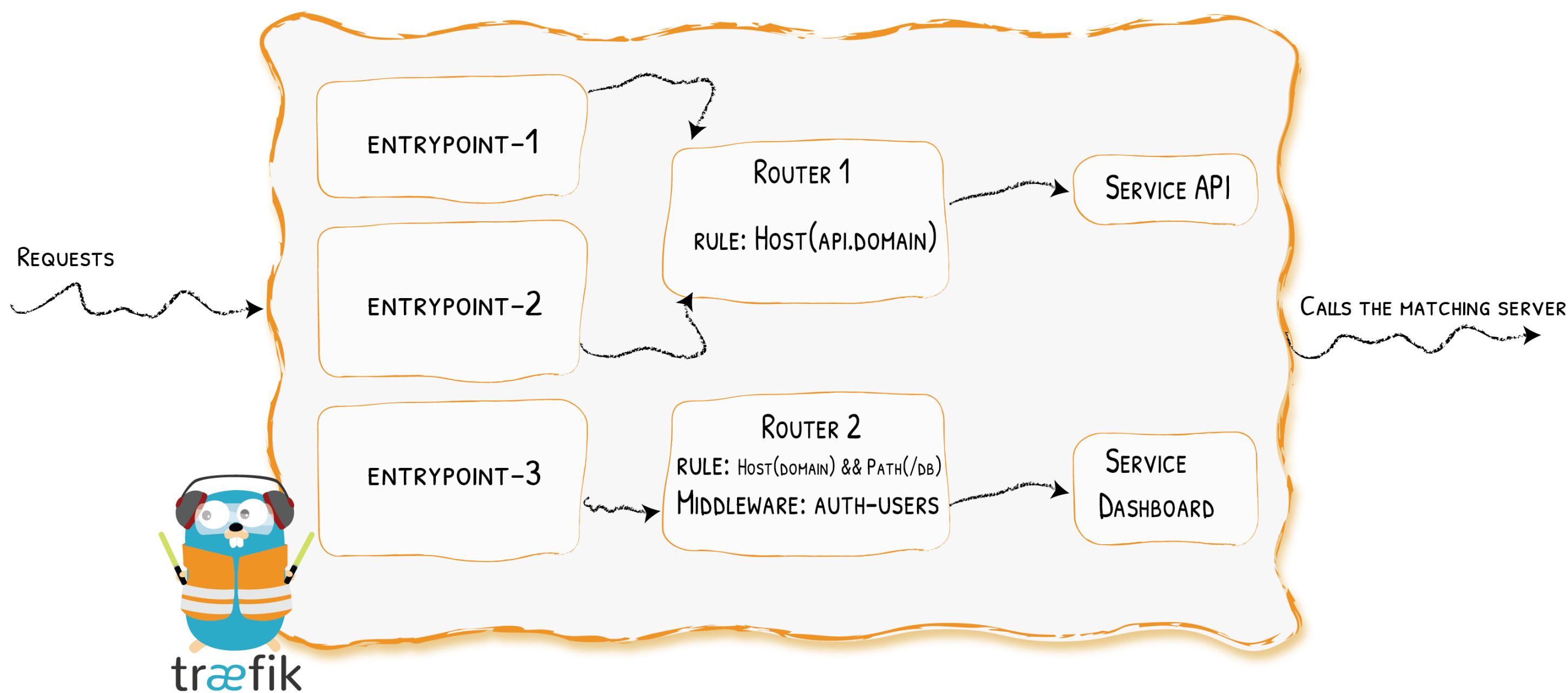
Architecture (V2.0) At A Glance



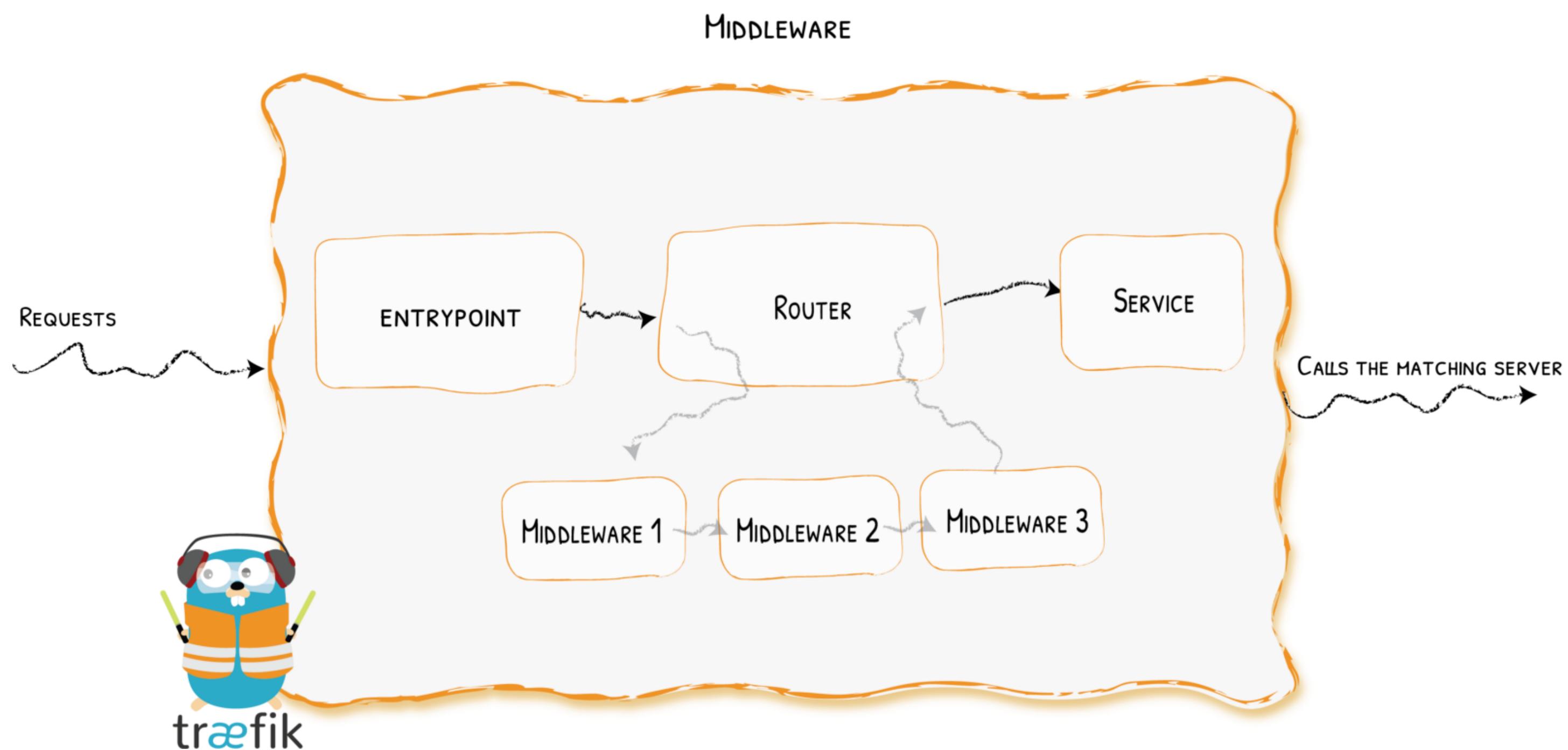
Entrypoints



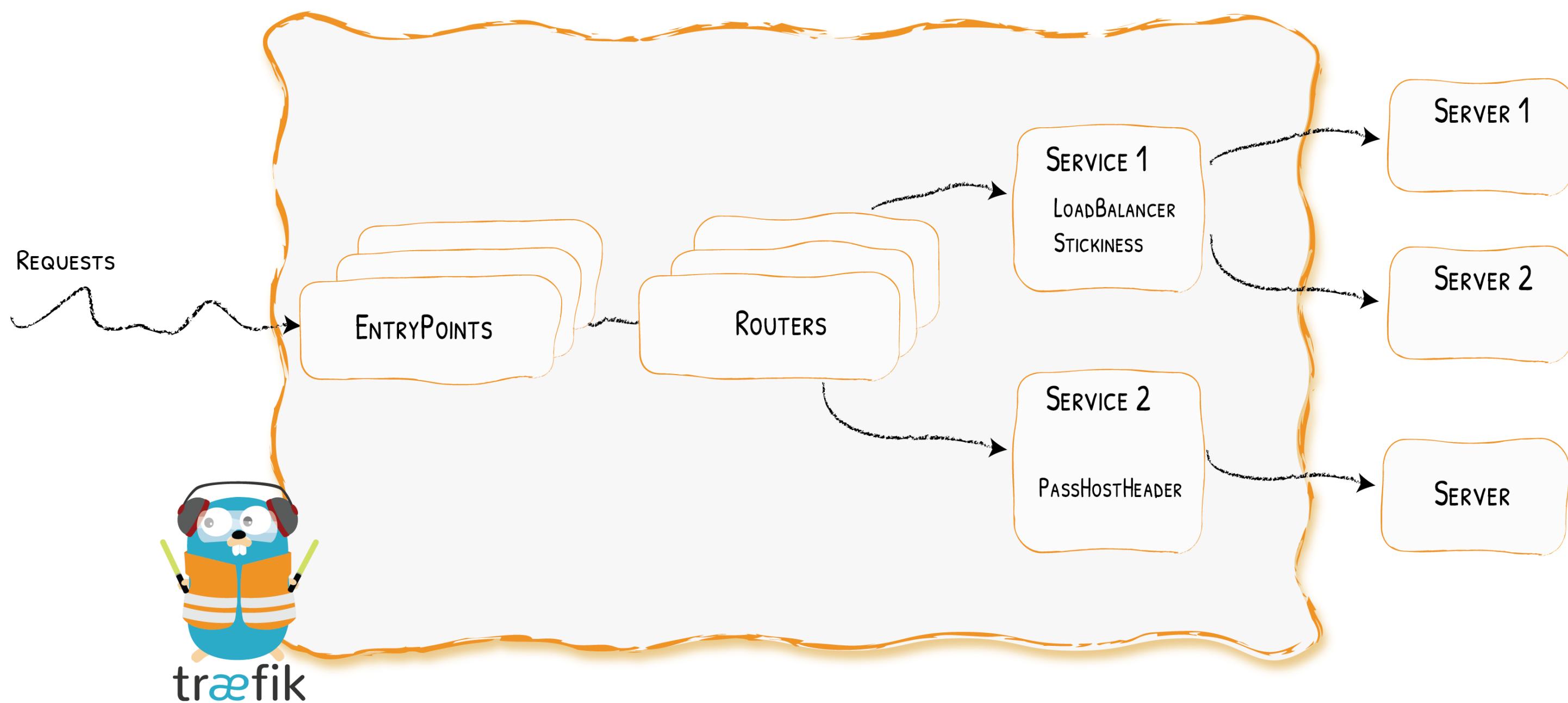
Routers



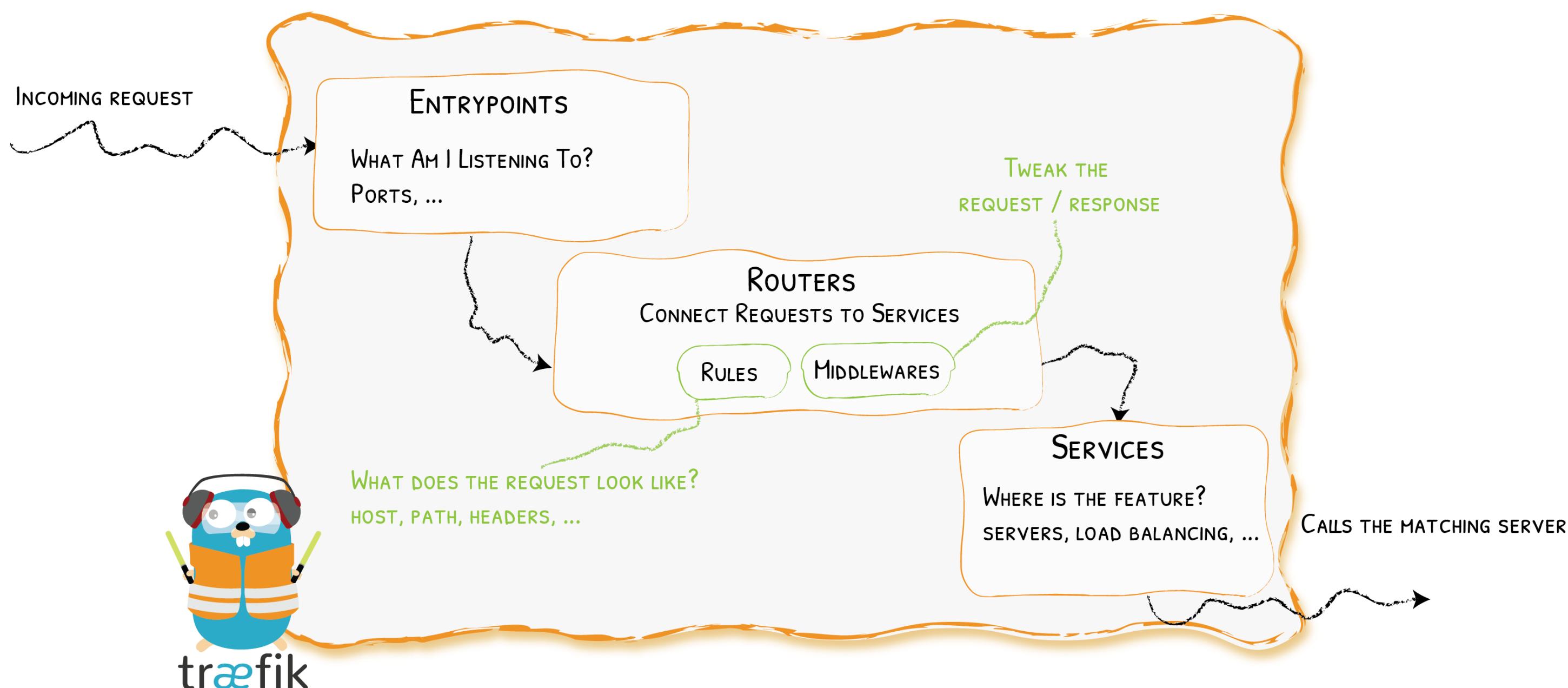
Middlewares



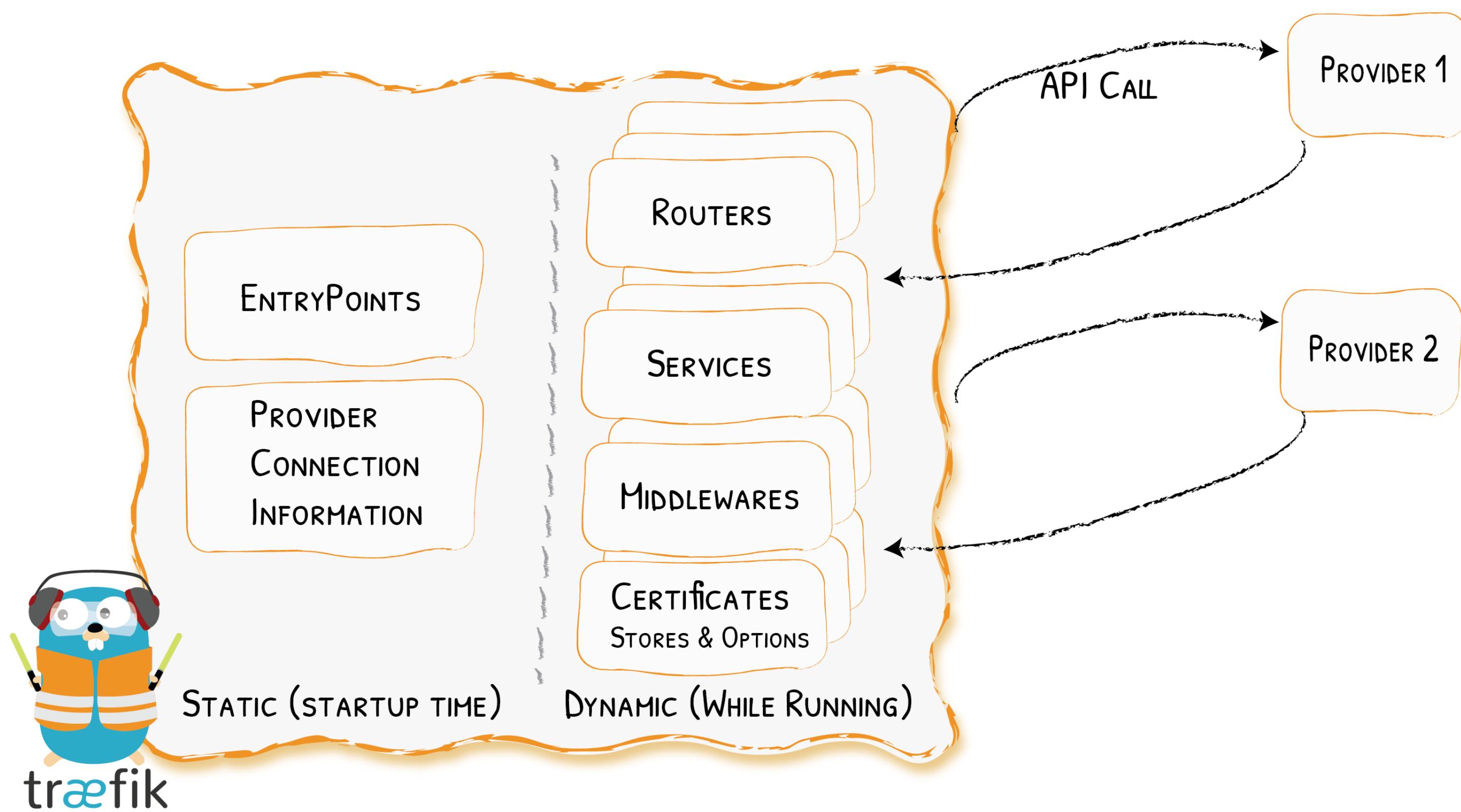
Services



Architecture (Again) At A Glance

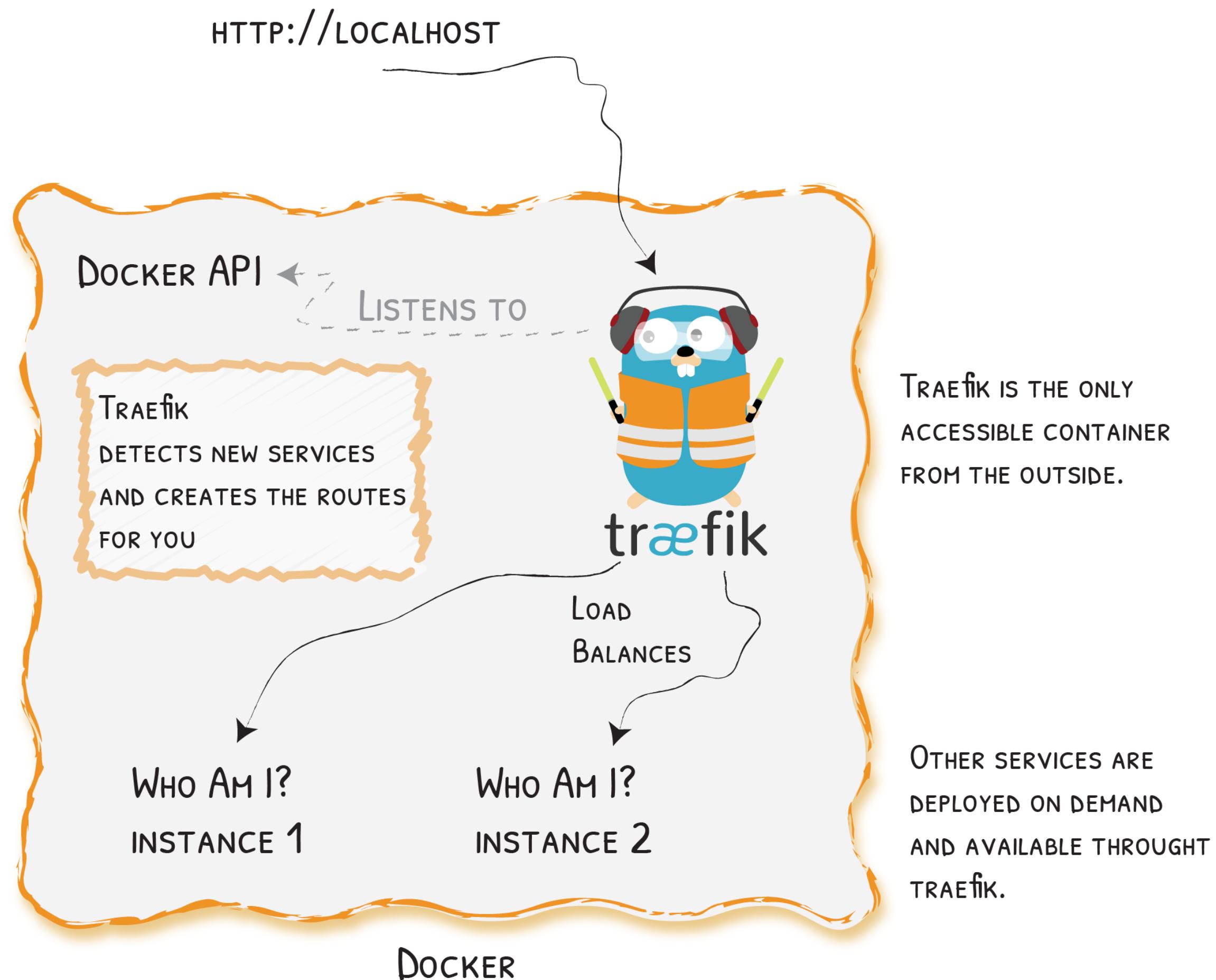


Static & Dynamic Configuration



Show Me The Configuration!

Simple Example With



With

- With Docker Compose:

```
version: '3'

services:
  reverse-proxy:
    image: traefik:v2.0
    command: --providers.docker
    ports:
      - "80:80"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock

  webapp:
    image: containous/whoami
    labels:
      - "traefik.http.routers.webapp.rule=Host(`localhost`)"
```

With 🐳: Context

```
# https://mycompany.org/jenkins -> http://jenkins:8080/jenkins
jenkins:
  image: jenkins/jenkins:lts
  environment:
    - JENKINS_OPTS=--prefix=/jenkins
  labels:
    - "traefik.http.services.jenkins.LoadBalancer.server.Port=8080" # Because 50000 is also exposed
    - "traefik.http.routers.jenkins.rule=Host(`mycompany.org`) && PathPrefix(`/jenkins`)"
    - "traefik.http.routers.jenkins.service=jenkins"
```

With : Rewrites

```
# https://mycompany.org/gitserver -> http://gitserver:3000/
gitserver:
  image: gitea/gitea
  labels:
    - "traefik.http.routers.gitserver.rule=Host(`mycompany.org`) && PathPrefix(`/gitsver`)"
    - "traefik.http.middlewares.gitserver-stripPrefix.stripPrefix.prefixes=/gitsver"
    - "traefik.http.routers.gitserver.middlewares=gitserver-stripPrefix"
```

With 🐳: Websockets

```
# https://webterminal.mycompany.org -> http://webterminal/
webterminal:
  image: ts10922/ttyd
  labels:
    - "traefik.http.routers.devbox.rule=Host(`webterminal.mycompany.org`)"
```

With File Configuration

Canary Releases

```
http:  
  services:  
    canary:  
      weighted:  
        services:  
          - name: appv1  
            weight: 3 # 75%  
          - name: appv2  
            weight: 1 #25%  
    appv1:  
      loadBalancer:  
        servers:  
          - url: "http://private-ip-server-1/"  
    appv2:  
      loadBalancer:  
        servers:  
          - url: "http://private-ip-server-2/"
```

Traefik With ⚓

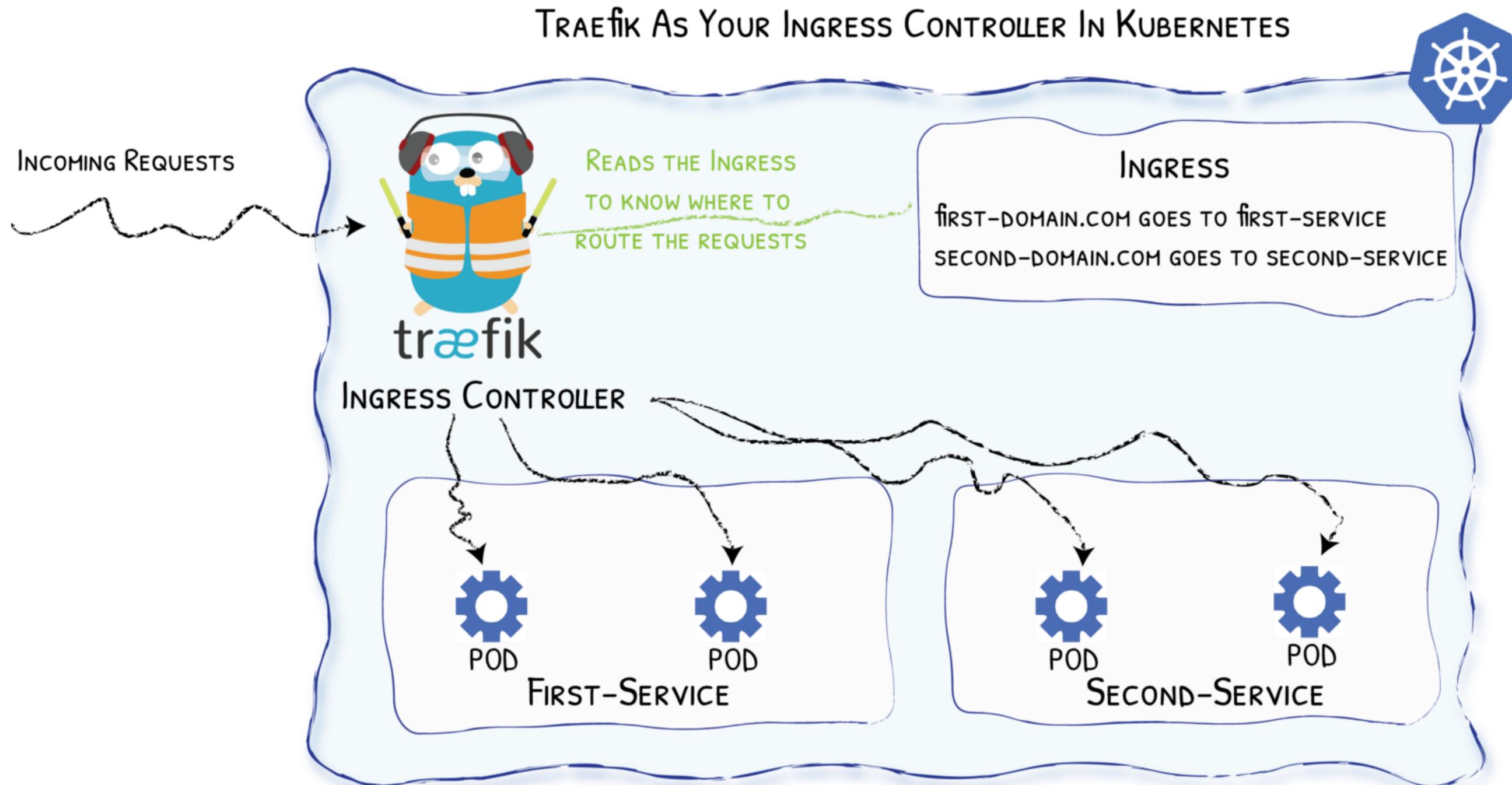


Diagram from <https://medium.com/@geraldcroes>

Example Code With ⚓

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: 'traefik'
spec:
  rules:
  - host: localhost
    http:
      paths:
      - path: "/whoami"
        backend:
          serviceName: webapp
          servicePort: 80
```

✳️ CRD - Custom Resources Definition

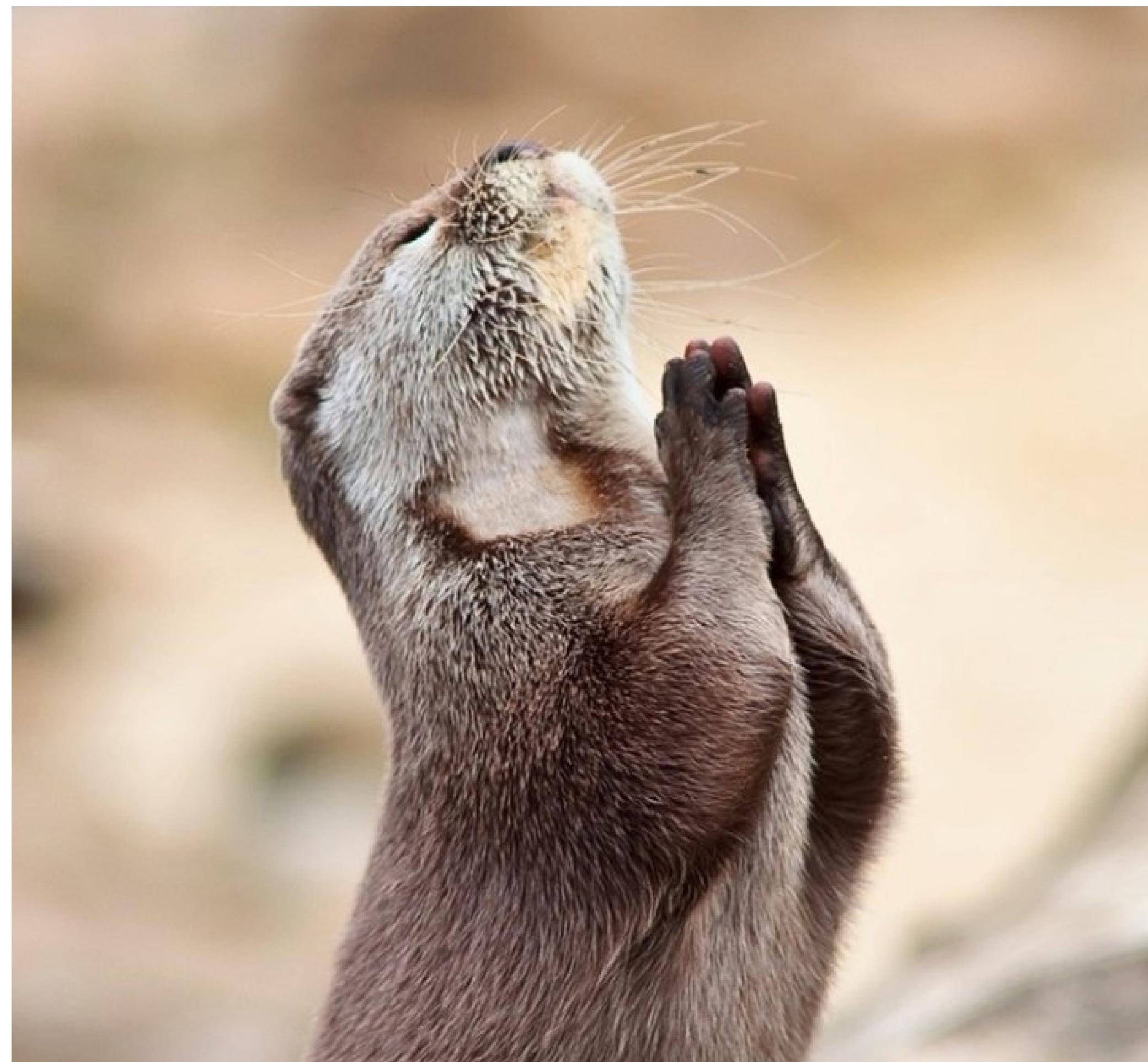
```
# File "webapp.yaml"
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: simpleingressroute
spec:
  entryPoints:
    - web
  routes:
    - match: Host(`localhost`) && PathPrefix(`/whoami`)
      kind: Rule
      services:
        - name: webapp
          port: 80
```

```
$ kubectl apply -f webapp.yaml
$ kubectl get ingressroute
```

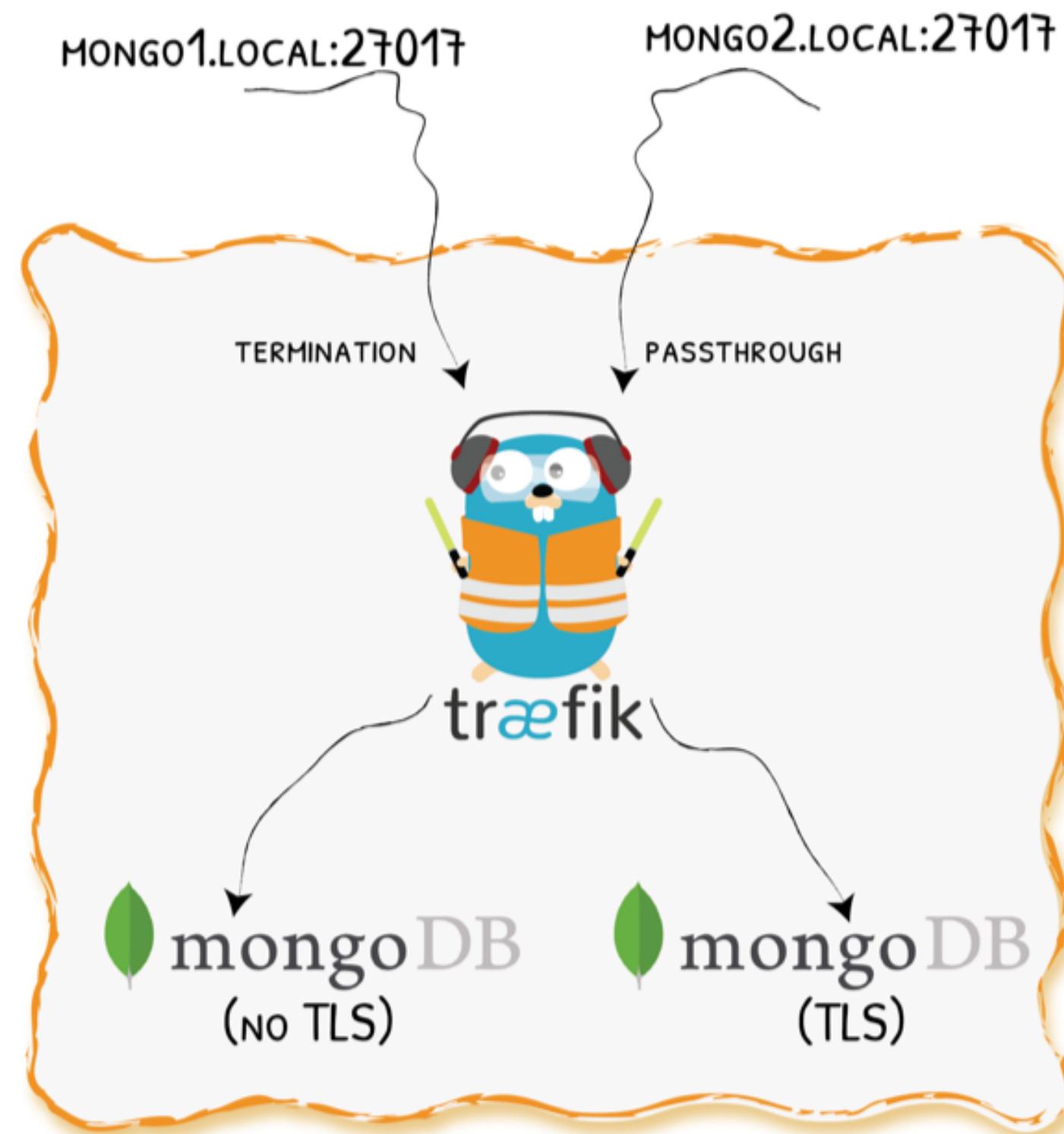
✳️ & TCP (With CRD)

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRouteTCP
metadata:
  name: ingressroutetcpmongo.crd
spec:
  entryPoints:
    - mongotcp
  routes:
    - match: HostSNI(`mongo-prod`)
      services:
        - name: mongo-prod
          port: 27017
```

Demo

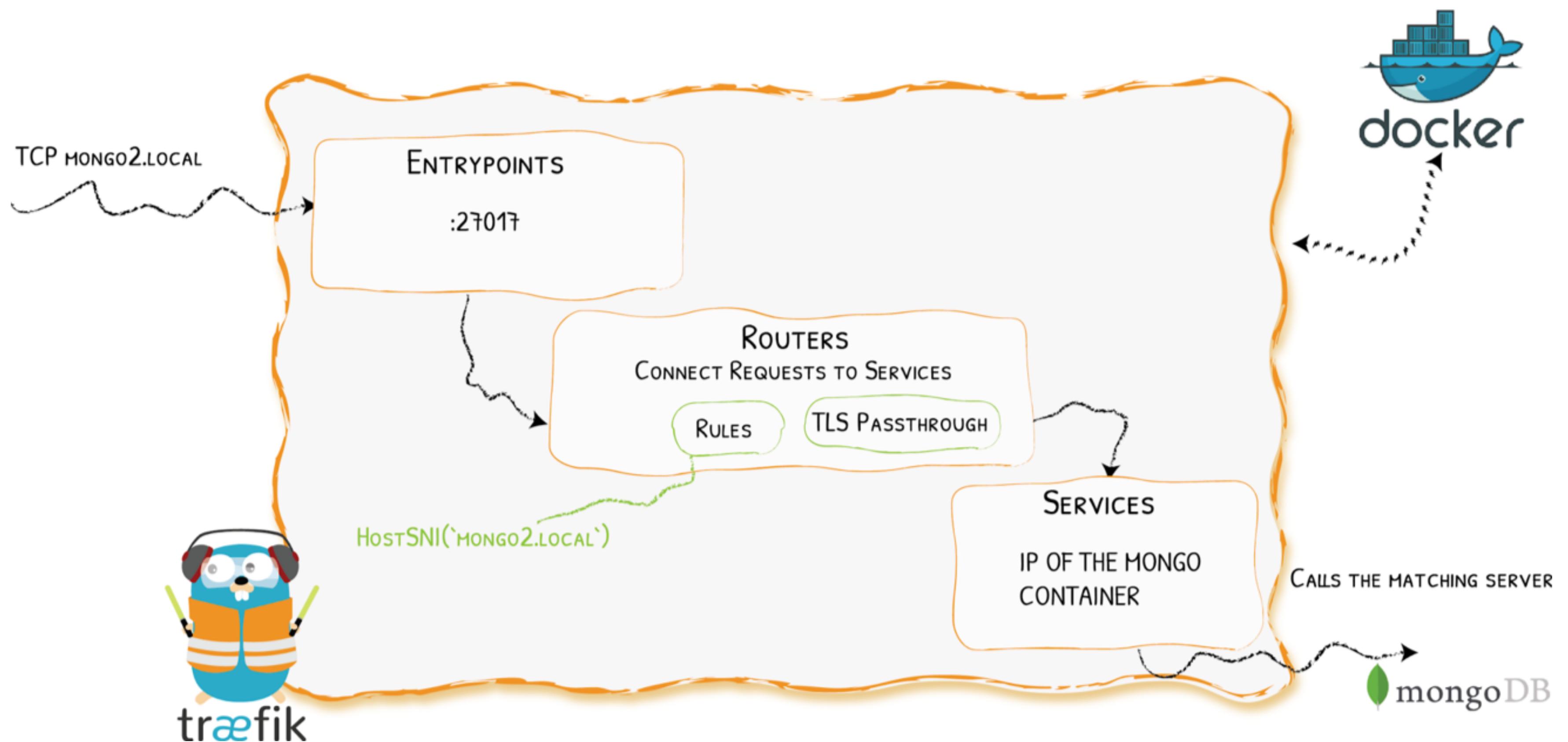


Demo 1 - SNI Routing + TLS Passthrough For TCP

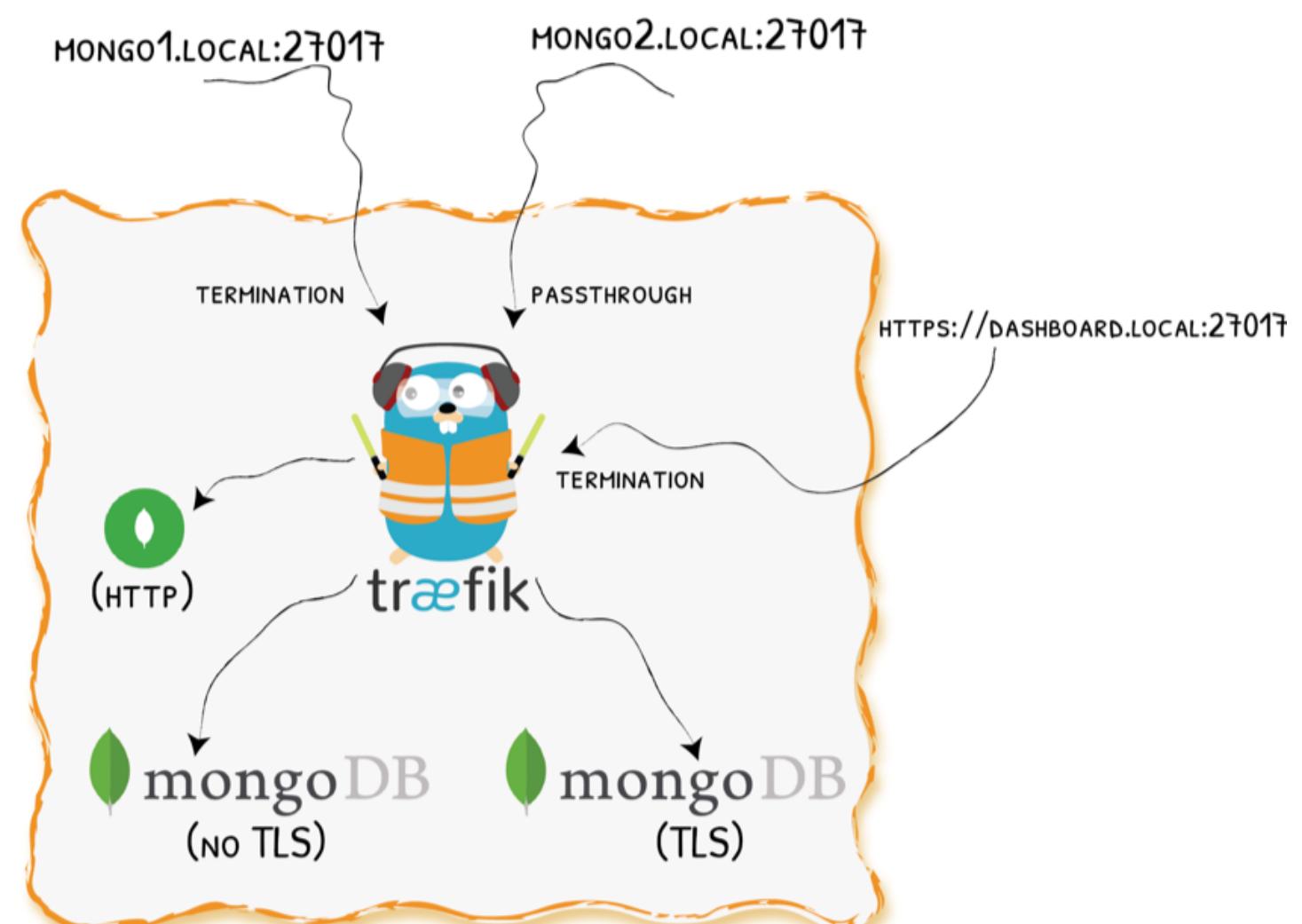


Demo Code on [GitHub](#)

Demo 1 - Configuration



Demo 2 - Muxing HTTPS And TCP On The Same Port



Demo Code on [GitHub](#)

Demo 3 - Canary Release Of A WebApp

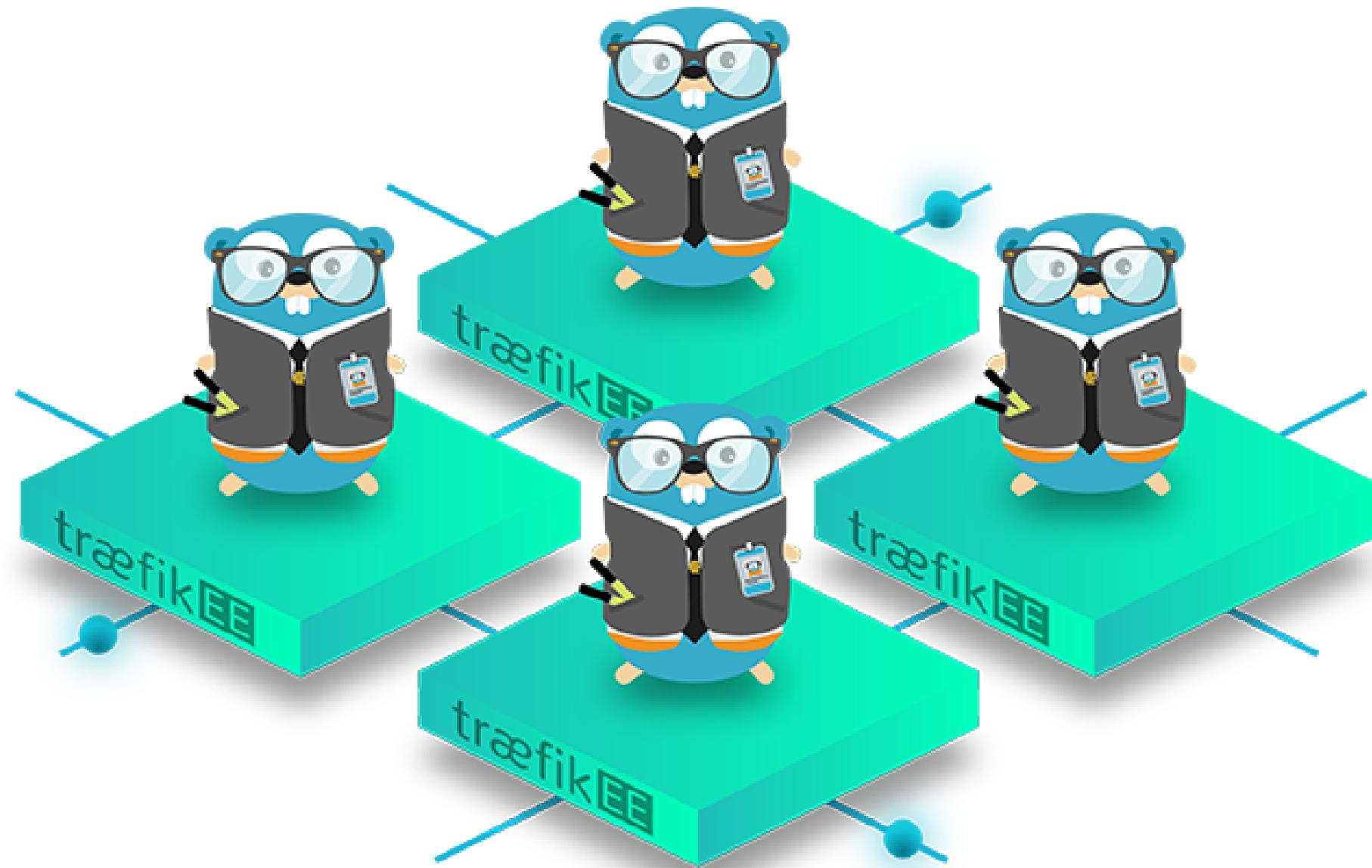
Demo 4 - Cheesy Kubernetes

The Herd



You came to the wrong neighbour

Traefik Comes In Herd



High Availability

HIGH AVAILABILITY

traefik ENTERPRISE EDITION



SECURITY

traefik ENTERPRISE EDITION

Scalability

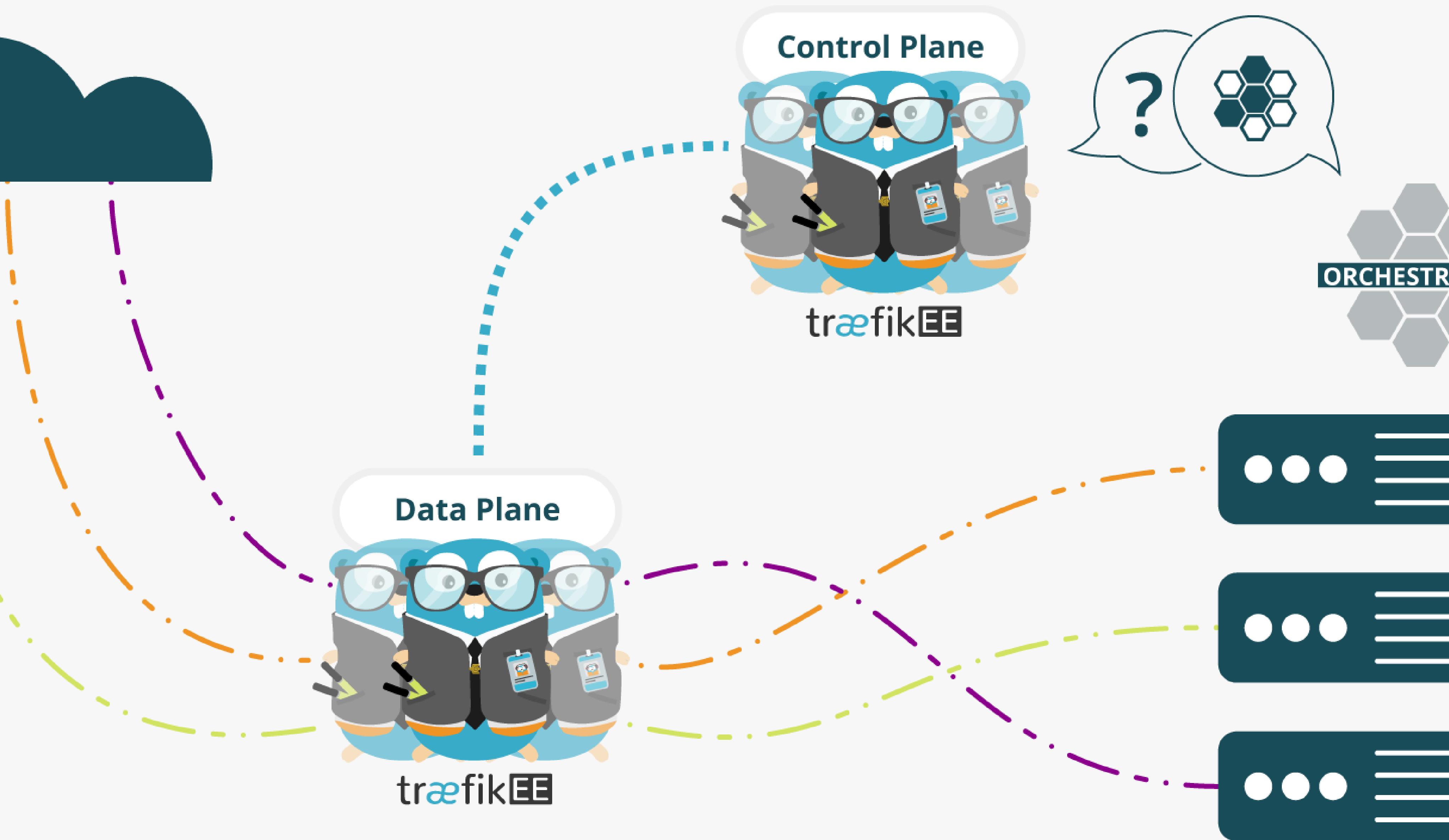


SCALABILITY

traefik ENTERPRISE EDITION

INTERNET

TO YOUR INFRA



As Simple As Traefik

- Install it:

```
# Cluster Installation
traefikeectl install \
  --licensekey="SuperSecretLicence" \
  --dashboard \
  --kubernetes # Or --swarm
```

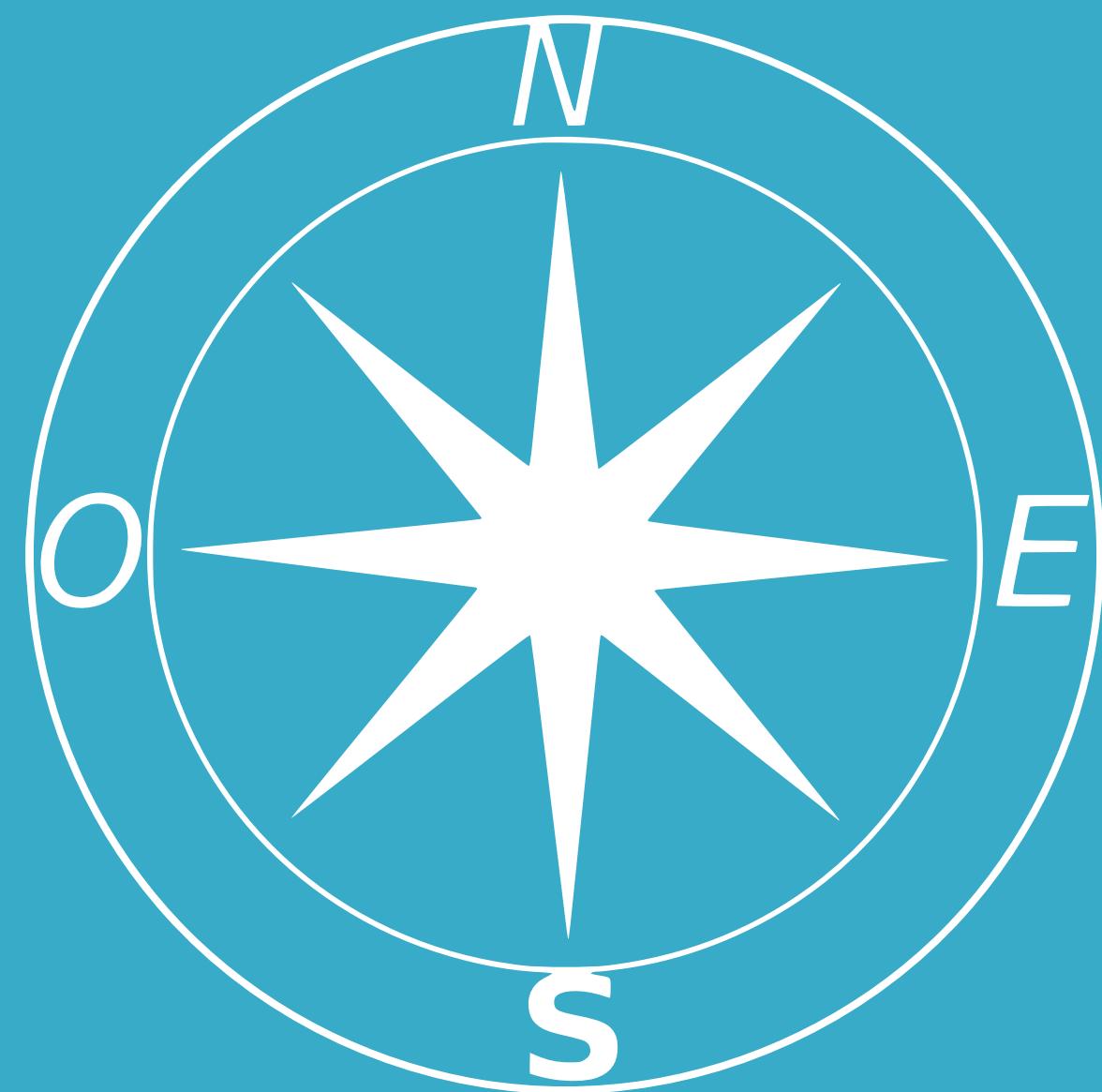
- Configure it:

```
# Routing Configuration, same as Traefik's
traefikeectl deploy \
  --acme.email=ssl-admin@mycompany.org
  --acme.tlsChallenge
  ...
```

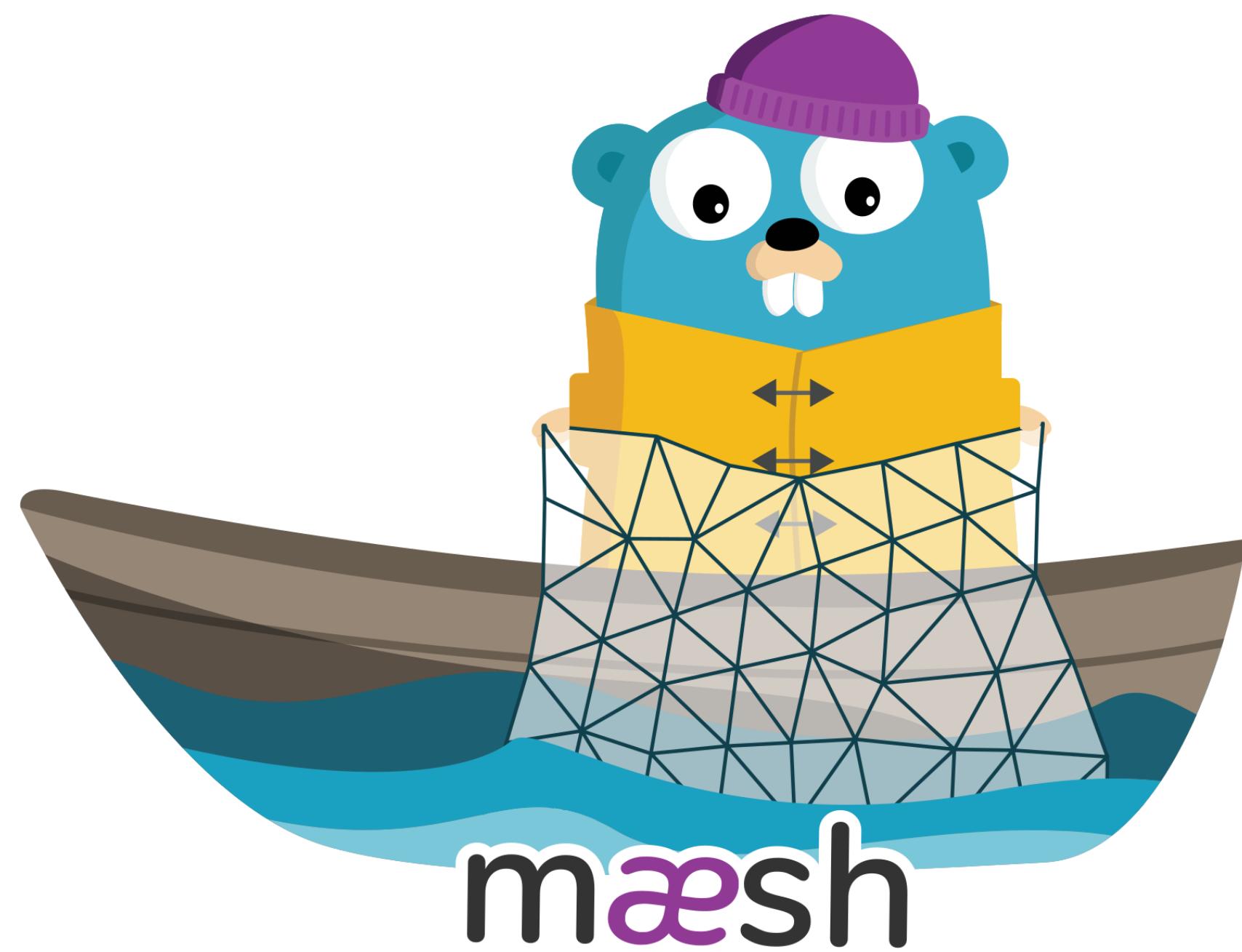
Free Trial

<https://containo.us/traefikee>

East / West Traefik



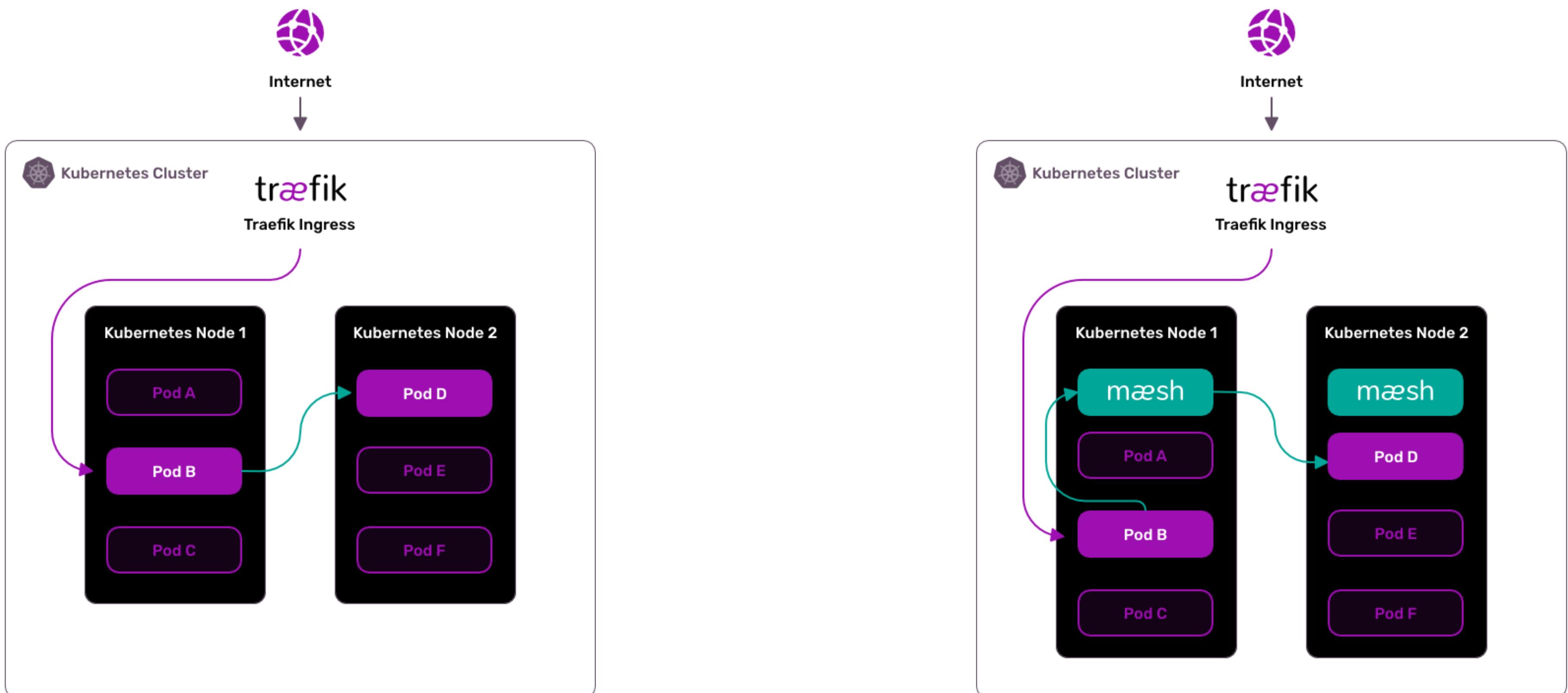
Say Hello To Maesh



What Is Maesh?

Maesh is a lightweight, easy to configure, and non-invasive service mesh that allows visibility and management of the traffic flows inside any Kubernetes cluster.

Maesh Architecture



More On Maesh

- Built on top of Traefik,
- SMI (Service Mesh Interface specification) compliant,
- Opt-in by default.

Maesh Website

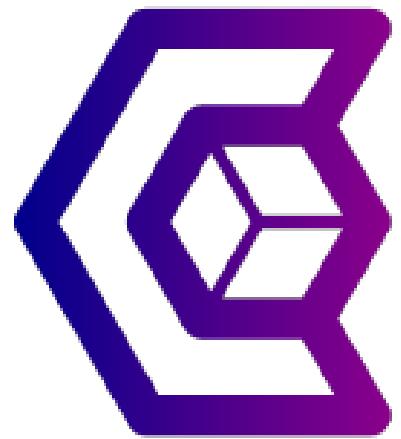
That's All Folks!



We Have
Stickers!

traefik

We Are Hiring!

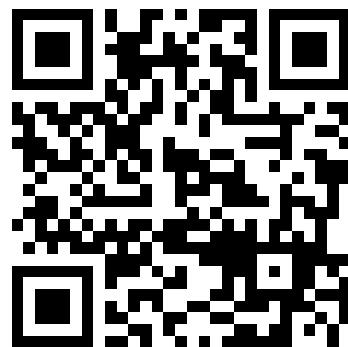


```
docker run -it containous/jobs
```

Thank You!

 @emilevauge

 emilevauge



- Slides (HTML): <https://containous.github.io/slides/kubernetes-meetup-sf>
- Slides (PDF): <https://containous.github.io/slides/kubernetes-meetup-sf/slides.pdf>
- Source on : <https://github.com/containous/slides/tree/kubernetes-meetup-sf>